

CORRECTED VERSION

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
20 July 2000 (20.07.2000)

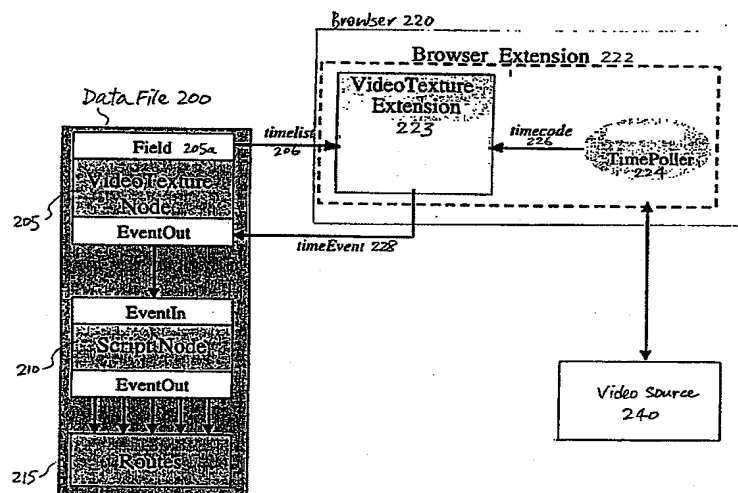
PCT

(10) International Publication Number  
WO 00/42773 A1

- (51) International Patent Classification<sup>7</sup>: H04N 7/26, 7/173
- (21) International Application Number: PCT/US00/01265
- (22) International Filing Date: 19 January 2000 (19.01.2000)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/116,453 19 January 1999 (19.01.1999) US  
09/484,492 18 January 2000 (18.01.2000) US
- (63) Related by continuation (CON) or continuation-in-part (CIP) to earlier application:  
US Not furnished (CON)  
Filed on 18 January 2000 (18.01.2000)
- (71) Applicant (for all designated States except US): SONY ELECTRONICS INC. [US/US]; 1 Sony Drive, Park Ridge, NJ 07656 (US).
- (72) Inventors; and
- (75) Inventors/Applicants (for US only): LE VAN GONG, Hubert [FR/US]; Apartment 8C, 444 Saratoga Avenue, Santa Clara, CA 95050 (US). RAFEY, Richter, A. [US/US]; 1394 Jackson Street, Santa Clara, CA 95050 (US). WANG, Sidney [US/US]; 2133 Pomezia Court, Pleasanton, CA 94566 (US). GIBBS, Simon [GB/US]; 1261 Mayberry Lane, San Jose, CA 95131 (US).
- (74) Agents: PALMER, John et al.; Ladas & Parry, Suite 2100, 5670 Wilshire Boulevard, Los Angeles, CA 90036-5679 (US).
- (81) Designated States (national): AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

[Continued on next page]

(54) Title: SYSTEM AND METHOD FOR IMPLEMENTING INTERACTIVE VIDEO



(57) Abstract: A system and method for interfacing graphics content with an independent video source to generate interactive media content wherein the source and the viewer can share control of the media content. In one embodiment, a computer implemented method for interfacing a 3-D graphics platform with broadcast video is provided. In this embodiment, the method comprises the step of defining a timelist comprising video triggers, wherein each of the video triggers represents a time at which an event is to occur within a 3-D graphics scene generated using the 3-D graphics platform. The method further comprises the step of receiving a timecode and a video frame from the broadcast video, wherein the timecode is associated with the video frame. The method further comprises the step of effectuating a behavior change for an object in the 3-D graphics scene in response to a match between one of the video triggers and the timecode such that the behavior change is synchronized with the video frame in real-time. In a preferred embodiment, the present invention includes the above and wherein the broadcast video comprises music video.

WO 00/42773 A1



**(84) Designated States (regional):** ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

**Published:**

— *With international search report.*

**(48) Date of publication of this corrected version:**

12 July 2001

**(15) Information about Corrections:**

see PCT Gazette No. 28/2001 of 12 July 2001, Section II

**Previous Correction:**

see PCT Gazette No. 44/2000 of 2 November 2000, Section II

*For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.*

## SYSTEM AND METHOD FOR IMPLEMENTING INTERACTIVE VIDEO

The present application claims priority to and is related to U.S. Patent Application No. 60/116,453 entitled "Interactive Music Video Based on 3D Computer Graphics and Broadcast Video" by Gibbs, et al. (filed on January 19, 1999) pursuant to 35 U.S.C. §119(e); 37 C.F.R. §1.78. This application is also a continuation of a U.S. patent application entitled "System and Method for Implementing Interactive Video Based on Three-Dimensional (3-D) Computer Graphics and Broadcast Video" by Gibbs, et al. and filed on January 18, 2000. (The serial number of the application filed on January 18, 2000 is not yet known.) U.S. Application No. 60/116,453 and the above-mentioned U.S. patent application filed on January 18, 2000 are incorporated herein by this reference.

BACKGROUND OF THE INVENTION15      FIELD OF THE INVENTION

The present invention relates to the design of interactive graphics and video systems. That is, the invention relates to a system and method for implementing interactive video based on three-dimensional (3-D) computer graphics and broadcast video. More specifically, the present invention pertains to a system and method for interfacing 3-D graphics content with an independent video source (e.g., broadcast video, etc.) to generate interactive media content.

RELATED ART

Traditional television broadcast has been a one-way communication channel. Until recently, virtually all available broadcast content is authored, edited and composited at the head end by content providers (who can either be the broadcasters or separate entities) such that all viewers have an identical view. Moreover, traditional televisions and other broadcast receivers typically do not have three-dimensional (3-D) graphics capability (e.g., specialized hardware and/or software) built into them. Thus, even though the notion of interactive television has been in existence for some time, due to bandwidth

limitations, the lack of 3-D graphics processing support and other reasons, interactive media content with 3-D graphics is not feasible or practicable in the traditional television broadcast paradigm.

5           The advent of digital television (DTV) technology has enabled the development of interactive content and its delivery to the viewers' homes. Generally, digital broadcast can be characterized as a high-speed data pipe into the home, providing dramatic bandwidth improvements over traditional broadcast for content delivery. Thus, once the digital broadcast infrastructure  
10 has been deployed, new types of applications, new kinds of services and new forms of entertainment become feasible. For example, this broadcast data pipe allows numerous forms of "enhanced television" programming (e.g., TV programs with accompanying data, such as game scores and statistics in a sports program) to be delivered to viewers at home, who enjoy wide latitudes to  
15 choose when and how to view the additional information. In addition, unlike traditional televisions and other broadcast receivers, industry-standard DTV receivers can be built to support local 3-D graphics acceleration. Therefore, it is also possible to develop sophisticated applications that use the high speed broadcast data pipe to incorporate interactive 3-D graphics into digital  
20 broadcast content to greatly enrich the viewers' experience.

          Since DTV technology can provide the requisite bandwidth for delivery of rich media content as well as the capability for processing 3-D graphics, next generation systems that support the integration of broadcast content and  
25 interactive 3-D graphics can be proposed, provided that a viable mechanism for interfacing the broadcast content and the graphics components is available. Thus, it would be highly advantageous to provide such an interfacing

mechanism to maximize the potential benefits afforded by the latest DTV technology.

Furthermore, it is appreciated that compatibility is essential in developing  
5 an interfacing mechanism. More specifically, numerous vendors will offer different appliances and applications for use in a DTV environment. As such, it would be desirable that these different appliances and applications can share a common interfacing mechanism such that they can work together seamlessly.

10 Additionally, it is appreciated that typical multimedia authoring tools are designed to operate within a self-contained environment and generally have a built-in runtime to verify the authored content. As such, these tools do not provide direct support for external, non-native interfacing mechanism. For example, lack of support for external broadcast triggering mechanism is  
15 prevalent in authoring tools for 3-D graphics platforms because such tools have traditionally not been considered applicable or useful in the context of television broadcasting. Thus, in an environment where broadcast content and interactive 3-D graphics are integrated, it would be desirable to provide a mechanism for authoring 3-D content in the context of broadcast triggering such that standard  
20 multimedia authoring tools can be used.

It is further realized that one particular type of interactive content that  
garners much interest is interactive music videos. Indeed, music videos have  
been a major element of the popular music industry since "MTV" came into  
25 existence in the early 1980s. More recently, as the "convergence" of television viewing and home computing accelerates, the notion of interactive music videos is being enthusiastically explored. Therefore, once a viable mechanism for

interfacing broadcast content and 3-D graphics components becomes available, it would be highly desirable to provide a method and system to deliver music videos as interactive content to viewers using DTV technology.

SUMMARY OF THE INVENTION

It would be advantageous to provide a mechanism for interfacing graphics content, particularly 3-D graphics, with broadcast video or other independent video so as to deliver interactive media content. Furthermore, it would also be  
5 advantageous for such interface to utilize an existing standard which has been adopted in the industry in its implementation such that the interface is widely compatible with other applications. Additionally, it would be highly desirable to utilize such an interface to provide interactive music video capability.

10 Accordingly, the present invention provides a system and method for interfacing graphics content with a video source to generate interactive media content wherein the video source (e.g., broadcaster, etc.) and the viewer can share control of the media content. By so doing, embodiments of the present invention provide greatly enhanced viewer experience over, for example, existing  
15 broadcast video programming. Moreover, embodiments of the present invention can be efficiently implemented within a standard 3-D graphics environment that supports interactivity. As such, the present invention leverages upon a versatile technology platform for 3-D graphics and delivers a system and method that is widely compatible with other applications. Moreover, embodiments of the present  
20 invention can be utilized to provide interactive music capability. These and other advantages of the present invention not specifically mentioned above will become clear within discussions of the present invention presented herein.

More specifically, in one embodiment of the present invention, a computer  
25 implemented method for interfacing a three-dimensional (3-D) graphics platform with broadcast video is provided. In this embodiment, the method comprises the step of defining a timelist comprising video triggers,

wherein each of the video triggers represents a time at which an event is to occur within a 3-D graphics scene generated using the 3-D graphics platform.

The method also comprises the step of accessing the timelist from a data file, which is an input format supported by the 3-D graphics platform. The method

5 further comprises the step of receiving a timecode and a video frame from the broadcast video, wherein the timecode is associated with the video frame.

Moreover, in this embodiment, the method comprises the step of comparing the video triggers and the timecode. Additionally, the method further comprises the step of effectuating a behavior change for an object in the 3-D graphics scene in

10 response to a match between one of the video triggers and the timecode such that the behavior change is synchronized with the video frame in real-time. In a specific embodiment, the present invention includes the above steps and wherein the 3-D graphics platform comprises a Virtual Reality Modeling Language (VRML) platform and the data file comprises a VRML scene

15 description file. In one embodiment, the present invention includes the above and wherein the timelist is stored in a VRML node of the VRML scene description file. In a preferred embodiment, the present invention includes the above and wherein the broadcast video comprises music video.

20 Embodiments of the present invention include the above steps and further comprise the step of embedding shaped video in the broadcast video, wherein the shaped video is partially transparent to provide special effects generated within the 3-D graphics scene. Additionally, embodiments of the present invention include the above and further comprises the steps of

25 transmitting feedback information to the source of the broadcast video and the source modifying contents of the broadcast video in response to the feedback information.



BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and form a part of this specification, illustrate embodiments of the invention and, together with  
5 the description, serve to explain the principles of the invention:

Figure 1A is an exemplary general purpose computer system with which  
embodiments of the present invention can be implemented.

10 Figure 1B is a block diagram illustrating an exemplary integrated  
broadcast and 3-D graphics environment in accordance with one embodiment  
of the present invention.

Figure 2 is a data flow diagram illustrating data flow for performing event  
15 triggering in accordance with one embodiment of the present invention.

Figure 3 is a flow diagram illustrating steps for interfacing a three-  
dimensional (3-D) graphics platform with broadcast video in accordance with  
one embodiment of the present invention.

20

Figure 4 is a flow diagram illustrating steps for implementing event  
triggering with a VRML browser in accordance with one embodiment of the  
present invention.

Figure 5 is a data flow diagram illustrating data flow for simulating event triggering in accordance with one embodiment of the present invention.

Figure 6 is a flow diagram illustrating steps for testing event triggering  
5 using simulated timecodes in accordance with one embodiment of the present invention.

### DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of the present invention, a system and method for implementing interactive video based on three-dimensional graphics and broadcast video, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be recognized by one skilled in the art that the present invention may be practiced without these specific details or with equivalents thereof. In other instances, well known methods, procedures, components, and circuits have not been described in detail as not to unnecessarily obscure aspects of the present invention.

### NOTATION AND NOMENCLATURE

Some portions of the detailed descriptions which follow are presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations of operations on data bits within a computer memory. These descriptions and representations are the means used by those skilled in the data processing arts to most effectively convey the substance of their work to others skilled in the art. A procedure, computer executed step, logic block, process, etc., is here, and generally, conceived to be a self-consistent sequence of steps or instructions leading to a desired result. The steps are those requiring physical manipulations of physical quantities. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated in a computer system. It has proven convenient at times,

principally for reasons of common usage, to refer to these signals as bits, values, elements, symbols, characters, terms, numbers, or the like.

It should be borne in mind, however, that all of these and similar terms  
5 are to be associated with the appropriate physical quantities and are merely  
convenient labels applied to these quantities. Unless specifically stated  
otherwise as apparent from the following discussions, it is appreciated that  
throughout the present invention, discussions utilizing terms such as "defining",  
"accessing", "receiving", "comparing", "effectuating" or the like, refer to the action  
10 and processes of a computer system (e.g., Figure 1A), or similar electronic  
computing device, that manipulates and transforms data represented as  
physical (electronic) quantities within the computer system's registers and  
memories into other data similarly represented as physical quantities within the  
computer system memories or registers or other such information storage,  
15 transmission or display devices.

Aspects of the present invention, described below, are discussed in  
terms of steps executed on a computer system. These steps (e.g., process 300)  
are implemented as program code stored in computer readable memory units of  
20 a computer system and are executed by the processor of the computer system.  
Although a variety of different computer systems can be used with the present  
invention, an exemplary general purpose computer system 100 is shown in  
Figure 1A.

### COMPUTER SYSTEM ENVIRONMENT

In general, as illustrated in Figure 1A, computer system 100 includes an address/data bus 102 for communicating information, a central processor 104 coupled to bus 102 for processing information and instructions, a volatile  
5 memory 106 (e.g., random access memory RAM) coupled to bus 102 for storing information and instructions for central processor 104 and a non-volatile memory 108 (e.g., read only memory ROM) coupled to bus 102 for storing static information and instructions for processor 104. It is appreciated that computer system 100 of Figure 1A is exemplary only and that the present invention can  
10 operate within a number of different computer systems including general purpose computer systems, embedded computer systems, and stand-alone computer systems specially adapted for video and/or graphics applications.

Computer system 100 also includes a data storage device 110 ("disk  
15 subsystem") such as a magnetic or optical disk and disk drive coupled with bus 102 for storing information and instructions. Data storage device 110 can include one or more removable magnetic or optical storage media (e.g., diskettes, tapes) which are computer readable memories. In accordance with the present invention, data storage device 110 can contain video and graphics  
20 data. Memory units of system 100 include 106, 108 and 110. Computer system 100 can also include a signal input output communication device 112 (e.g., modem, network interface card NIC, serial digital input) coupled to bus 102 for interfacing with other computer systems and/or data sources. In accordance with the present invention, signal input output communication device 112 can  
25 receive various incoming media streams (e.g., video signals).

Also included in computer system 100 of Figure 1A is an optional alphanumeric input device 114 including alphanumeric and function keys coupled to bus 102 for communicating information and command selections to central processor 104. Computer system 100 also includes an optional cursor control or directing device 116 coupled to bus 102 for communicating user input information and command selections to central processor 104. An optional display device 118 can also be coupled to bus 102 for displaying information to the computer user. Display device 118 may be a liquid crystal device (LCD), other flat panel display, cathode ray tube (CRT), or other display device suitable for creating graphic images and alphanumeric characters recognizable to the user. Cursor control device 116 allows the computer user to dynamically signal the two dimensional movement of a visible symbol (cursor) on a display screen of display device 118. Many implementations of cursor control device 116 are known in the art including a trackball, mouse, touch pad, joystick or special keys on alphanumeric input device 114 capable of signaling movement of a given direction or manner of displacement. Alternatively, it will be appreciated that a cursor can be directed and/or activated via input from alphanumeric input device 114 using special keys and key sequence commands. The present invention is also well suited to directing a cursor by other means such as, for example, voice commands.

It is appreciated that computer system 100 described herein illustrates an exemplary configuration of an operational platform upon which embodiments of the present invention can be implemented. Nevertheless, other computer

systems with differing configurations can also be used in place of computer system 100 within the scope of the present invention.

## INTEGRATED BROADCAST VIDEO AND 3-D GRAPHICS ENVIRONMENT

### 5                   IN ACCORDANCE WITH THE PRESENT INVENTION

Referring next to Figure 1B, a block diagram illustrating an exemplary integrated broadcast and 3-D graphics environment 150 in accordance with one embodiment of the present invention is shown. As illustrated in Figure 1B, within DTV environment 150, set top boxes (STBs) 151, 152 and 153 receive  
10 broadcast media streams 168 from a broadcast source 160. In one embodiment, STBs 151, 152 and 153 each comprises computer system 100 of Figure 1A. In a preferred embodiment, STBs 151, 152 and 153 are DTV receivers having built-in 3-D graphics processing capability and broadcast media streams 168 can include a combination of audio streams, video streams,  
15 3-D graphics streams and event trigger streams (e.g., tve-triggers under the ATVEF standard, described below). It is appreciated that the high bandwidth data channel provided by DTV environment 150 enables additional media elements such as 3-D graphics and interface components and event triggers to be broadcast along with traditional audio and video streams. Furthermore, it is  
20 appreciated that media local objects (e.g., graphics objects) in DTV receivers 151, 152 and 153 can leverage triggering mechanisms associated with the broadcast, thereby invoking behaviors that are synchronized with the broadcast.

Moreover, a DTV environment is also conducive to integration with the  
25 Internet 170, which can be used as an additional broadcasting source for data

and media objects and as a feedback channel for bi-directional communication. As such, a complete communication loop among the viewers and the broadcaster (e.g., broadcast source 160) can be established. Thus, referring still to Figure 1B, any of STBs 151, 152 and 153 can be coupled to a server over the Internet 170 within DTV environment 150. In an illustrative embodiment shown in Figure 1B, STB 151 is coupled to a virtual environment server (VES) 180 over the Internet 170 (e.g., via a "back channel"). In this embodiment, VES 180 supports a "virtual world" comprising various media objects each of which has its own set of attributes. Certain of these attributes, such as the media object's position on the display, can be controlled by the viewer. Based upon viewer actions that affect the attributes of the media objects, VES 180 updates the state of this virtual world. VES 180 also communicates with broadcast source 160 (e.g., over the Internet 170) in an embodiment as shown in Figure 1B. In one embodiment, broadcast source 160 receives feedback from the viewers (e.g., from STBs 151, 152 and 153 and through VES 180) and reacts (e.g., modifies the contents based on the feedback) accordingly, thereby allowing a high degree of personalized content distribution.

It is appreciated that in order to fully implement an integrated broadcast video and 3-D graphics environment 150 as shown in Figure 1B for providing the functionalities as described above, several technology components are necessary. First of all, an effective technique for blending broadcast video into a 3-D virtual environment (e.g., interactive 3-D graphics scenes) is needed for seamless integration of video and 3-D graphics in environment 150. Moreover, a mechanism for event triggering via broadcast media streams is also



necessary for implementing enhanced content. Additionally, a mechanism for streaming dynamic elements into content is needed to enable the injection of real-time effects into the integrated video and graphics scene. Furthermore, incorporating multi-user technology into environment 150 can facilitate broad viewer participation by allowing different viewers, celebrity characters and program hosts to interact in a shared virtual world such as environment 150.

Virtual Reality Modeling Language (VRML) Extensions in accordance with the Present Invention

10 In a currently preferred embodiment, the present invention provides a set of extensions to the Virtual Reality Modeling Language (VRML) to enable video from a live broadcast to appear in an animated 3-D scene associated with the video content. It is appreciated that VRML is an International Standards Organization (ISO) standard for 3-D graphics on the Internet. Furthermore, it is  
15 also appreciated that VRML is being included as the 3-D scene representation in a standard called MPEG-4 proposed by the Motion Picture Expert Group (MPEG). Indeed, VRML is being fully implemented in STBs by some vendors. As such, VRML is an ideal platform upon which 3-D graphics functionality can be integrated into next generation set-top box technologies. Thus, by extending  
20 VRML to implement broadcast video and 3-D graphics integration, the present invention leverages upon a versatile technology platform for 3-D graphics and delivers a system and method that is widely compatible with other applications.

More specifically, in one embodiment, the VRML extensions of the  
25 present invention comprise a new node definition. It is appreciated that nodes

in VRML can be given arbitrary names (e.g., via the DEF construct) and that it is easy to associate value changes in different VRML nodes provided that the data types involved are compatible. An exemplary semantic definition of the new node in accordance with the present invention is shown as follows:

```
5      VideoTexture {  
          field      SFString    source      "SDI"  
          field      SFCOLOR    chromaKey  0 0 0  
          field      MFInt32     timelist    []  
          eventOut   MFInt32     timeEvent  
10     }
```

In this embodiment, the source field of the VideoTexture node indicates the source from which the VRML browser of the present invention is receiving video input. Within the scope of the present invention, the value of the source field is hardware-dependent. In one embodiment, the source field can have one of two values, namely, the serial digital input (SDI) and the Ethernet port. In this embodiment, a value of SDI in the source field indicates that the video input is coming directly from an external digital video tape recorder/player (VTR). On the other hand, a value of Ethernet in the source field means that the video input is packetized and received over the Ethernet. An embodiment utilizing the Ethernet delivery mechanism is described in greater detail further below.

Moreover, the VideoTexture extension of the present invention supports both plain video and "shaped video". With reference again to the semantic definition of the VideoTexture node above, in one embodiment, the chromaKey

field stores chromakeying color information, which enables the handling of shaped or masked video as described below. Furthermore, the timelist field contains a list of quadruples. In one embodiment, each quadruple (h, m, s, f) represents a timestamp at which an event is expected to occur, where h stands for hour, m for minute, s for second, and f for frame. Importantly, in a preferred embodiment, the timelist field is used to facilitate event triggering in a VRML scene. The event triggering mechanism is described in greater detail below with reference to Figures 2 and 3. Additionally, in one embodiment, if an event does occur, the corresponding timestamp is returned via the eventOut named timeEvent.

#### Video Texturing and Masking in accordance with the Present Invention

Within the scope of the present invention, video texture mapping is an effective technique for blending broadcast video into a 3-D virtual environment. It is appreciated that texture mapping in general is a well-known technique to one of ordinary skill in the art. In one embodiment of the present invention, the video texture appears as a simple, flat "video wall" in the 3-D scene. In another embodiment, the video texture is mapped onto surfaces of more complex geometry than a flat surface, in the same fashion as texture mapping is performed in a typical 3-D graphics application.

Moreover, within the scope of the present invention, the texture source can be any video device, such as a video tape recorder/player (VTR) or a digital video disk player (DVD), or a live camera feed in the case of broadcast video. In one embodiment, an SGI Octane™ graphics workstation from Silicon

Graphics, Inc. of Mountain View, California, is used to implement video texture mapping. The Octane supports full-frame-rate video textures. More specifically, in this embodiment of the present invention, video streams are captured directly into texture memory. Importantly, once captured, a video field can be used as a texture as if it was an image loaded into the texture memory. Furthermore, in one embodiment, two texture buffers are used to enable double-buffering. More particularly, when a video field is being captured into one of the texture buffers, the other texture buffer can be used for drawing (e.g., rendering). Significantly, this overlapping of the video field loading process and the drawing process by using dual texture buffers enables real-time video texturing.

Additionally, in one embodiment, special effects called "shaped video" can be implemented using the VRML extensions of the present invention. More specifically, "shaped video" refers to video footage that can be made partially transparent to enable special composition effects. It is appreciated that the concept of "shaped video" is being addressed in the MPEG-4 standard. By providing the VRML extensions comprising the VideoTexture node as described above, the present invention enables "shaped video" to be efficiently implemented within a standard distributed 3-D graphics platform (e.g., VRML) that supports interactivity.

In one embodiment, the present invention explicitly transmits a mask for the video as part of the video signal in order to implement shaped video effects. In another embodiment, a chromakey in the video image (e.g., chromaKey field of the VideoTexture node) is used to define a mask for generating shaped video

effects. A rendering engine at the viewer's end (e.g., STBs 151, 152 and 153 of Figure 1B) then makes the appropriate region(s) of the video transparent (e.g., visible to the viewer) as specified by the mask. With this video masking technique, the rendering operation generates arbitrarily shaped video objects  
5 irrespective of the actual shape of the target object.

#### Event Triggering in accordance with the Present Invention

It is appreciated that industry-standard VRML (e.g., VRML97) has built-in mechanism for generating and responding to events. On the other hand, a  
10 consortium of broadcast and cable networks, in collaboration with consumer electronics companies, has put forth the Advanced Television Enhancement Forum (ATVEF) specification with a goal to provide a standard for enhanced television programming. It is appreciated that the ATVEF specification is not limited to digital TV or broadcast-only environments. In particular, ATVEF  
15 defines the notion of a trigger, called tve-trigger, which are real-time events sent to television receivers as part of an enhanced TV program. In response to these triggers, the receivers react and perform certain actions to augment the program content. For example, when an event trigger is received, a receiver can start a local script and/or inform the viewer that enhanced content has arrived. It is  
20 appreciated that while the functionalities that an ATVEF-compliant receiver should support are well-defined in the specification, the exact manner in which an event trigger is handled by a receiver can vary with implementations. Thus, the ATVEF specification and other efforts in this area address the definition of data channel and triggering standards and provide a robust mechanism for  
25 synchronized event delivery.

Based upon the framework (e.g., data channel, triggering standards, synchronized event delivery) defined in the ATVEF specification and the built-in capability (e.g., event generation, response to event) of VRML, the present

5 invention provides a novel mechanism in VRML for registering external events to track in the broadcast data stream. One embodiment of the present invention implements this mechanism by abstracting broadcast trigger events in a newly defined VRML node, namely, the VideoTexture node as described above. In one embodiment, the VRML node for tracking external events is implemented

10 as a numeric registry of time codes, so that only those events that are registered will cause event propagation in the VRML scene. Moreover, in this embodiment, VRML also represents connections between objects in the 3-D scene and has built-in animation mechanisms. As such, high level authoring of media events based on broadcast triggers is feasible once the interface

15 between the broadcast channel and VRML has been defined. These aspects of the present invention are described in greater detail below with reference to Figures 2 and 3.

Referring next to Figure 2, a data flow diagram illustrating data flow for

20 performing event triggering in accordance with one embodiment of the present invention is shown. As depicted in Figure 2, data is stored in a data file 200 authored by a content creator. In a currently preferred embodiment, data file 200 is a VRML scene description file having a VideoTexture node 205 of the present invention as described above, a script node 210 as well as routes 215.

25 Script nodes and routes are elements of VRML and are known to those of

ordinary skill in the art. Data in VRML scene description file 200 includes a timelist 206, which is a list of times at which triggering events are expected to occur. In one embodiment, timelist 206 is stored in the timelist field 205a of VideoTexture node 205.

5

Referring still to Figure 2, a browser 220 is used to read and process data from data file 200. In a currently preferred embodiment, browser 220 is a VRML browser and includes a browser extension 222, which in turn comprises a VideoTexture extension 223 and a time poller 224. In one embodiment, time  
10 poller 224 extracts a current timecode 226 from a video source 240 via a serial port during each frame rendering cycle. In another embodiment, timecode 226 is embedded in a video signal sent to browser 220 from video source 240. Furthermore, in one embodiment, video source 240 comprises video signals from a VTR. In another embodiment, video source 240 comprises a live video  
15 feed (e.g., broadcast video signals).

Within the scope of the present invention, VideoTexture extension 223 receives timelist 206 from VideoTexture node 205 and timecode 226 from time poller 224. VideoTexture extension 223 then compares the data in timelist 206  
20 against the current timecode 226. When a match between an item (e.g., a timestamp) in timelist 206 and current timecode 226 is detected, corresponding time event 228 of VideoTexture node 205 is sent to script node 210 for further processing. In one embodiment, script node 210 serves to launch one or more routes (e.g., routes 215) within the VRML scene when time event 228 is  
25 received from VideoTexture extension 223. Routes 215 in turn trigger behavior

changes of VRML objects in the scene. Moreover, it is appreciated that the content author is responsible for ensuring that there exists a one-to-one correspondence between the times listed in timelist 206 of VideoTexture node 205 and the time events listed in script node 210. In one embodiment, the present invention provides an authoring tool that is tailored to facilitate the authoring and editing of VRML scene description file 200, especially with respect to timelist field 205a of VideoTexture node 205 and events in script node 210.

10 Referring next to Figure 3, a flow diagram illustrating steps for interfacing a three-dimensional (3-D) graphics platform with broadcast video in accordance with one embodiment of the present invention is shown. In step 310, a timelist comprising video triggers is defined. In one embodiment, each of the video triggers represents a time at which an event is to occur within a 3-D graphics scene generated using the 3-D graphics platform of the present invention.

Referring still to Figure 3, in step 320, the timelist defined in step 310 is received from a data file for processing. In one embodiment, the data file is in an input format supported by the 3-D graphics platform of the present invention.

20 In one embodiment, the timelist comprises timelist 206 of VideoTexture node 205 and the data file comprises VRML scene description file 200, both of which are depicted in Figure 2.

With reference still to Figure 3, in step 330, a timecode and a video frame from a media stream of the broadcast video is received for processing, wherein



the timecode is characteristic of the video frame. In one embodiment, the timecode comprises timecode 226 of Figure 2.

Referring again to Figure 3, in step 340, the video triggers and the  
5 timecode are compared. In one embodiment, the comparison is performed using VideoTexture extension 223 of Figure 2.

With reference again to Figure 3, in step 350, when a match is detected between one of the video triggers and the timecode, a behavior change for an  
10 object in the 3-D graphics scene is effectuated according to the matched video trigger such that the behavior change is synchronized with the video frame in real-time.

Referring still to Figure 3, in step 360, shaped video is embedded in the  
15 media stream, wherein the shaped video is partially transparent such that special effects can be generated within the 3-D graphics scene.

Referring again to Figure 3, in step 370, input from a viewer is accepted. In one embodiment, contents of the broadcast video are capable of changing in  
20 response to the input. In another embodiment, the input can effectuate behavior change(s) for object(s) of the 3-D graphics scene.

With reference again to Figure 3, in step 380, feedback information is transmitted to the source of the broadcast video such that the source can modify  
25 contents of the broadcast video accordingly. In one embodiment, the feedback

information includes the viewer input described above in step 370. A method for interfacing a three-dimensional (3-D) graphics platform with broadcast video in accordance with embodiments of the present invention is thus described.

5 Referring next to Figure 4, a flow diagram illustrating steps for implementing event triggering with a VRML browser in accordance with one embodiment of the present invention is shown. In step 410, a list of times at which triggering events are expected to occur is received by the VRML browser of the present invention. In one embodiment, with reference back to Figure 2,  
10 timelist-206 of VideoTexture node 205 in VRML scene description file 200 is received by VideoTexture extension 223 of VRML browser 220. Moreover, in one embodiment, timelist 206 comprises a list of quadruples, wherein each quadruple (h, m, s, f) represents a timestamp at which an event is expected to occur, and wherein h stands for hour, m for minute, s for second, and f for frame.

15

Referring still to Figure 4, in step 420, a video frame is received from a video sub-system. In one embodiment, the video sub-system comprises video source 240 of Figure 2, which can provide video signals from a VTR or a live video feed as described above.

20

With reference still to Figure 4, in step 430, the received video frame is stored in a graphics texture memory. In one embodiment, the video frame is stored within texture memory (e.g., volatile memory 106 of Figure 1A). It is appreciated that once a video frame is stored in texture memory, the frame can  
25 be used as a texture as if it was a graphics image. As described above, in one

embodiment, two texture buffers are used to enable double-buffering. In this embodiment, when a video frame is being captured into one of the texture buffers, the other texture buffer can be used for drawing (e.g., rendering). Such dual-texture-buffer embodiment of the present invention thus enables real-time  
5 video texturing.

Referring again to Figure 4, in step 440, a timecode corresponding to the stored video frame is extracted from the video sub-system. In one embodiment, the timecode is extracted via a serial port during each frame rendering cycle.  
10 More specifically, in one embodiment, referring back to Figure 2, time poller 224 extracts timecode 226 from video source 240 and sends the extracted timecode 226 over to VideoTexture extension 223.

With reference again to Figure 4, in step 450, it is determined whether or  
15 not there is a match between an item in the timelist and the extracted timecode. In one embodiment, VideoTexture extension 223 compares the data in timelist 206 against the current timecode 226. If a match is detected, process 400 proceeds to step 460; otherwise, process 400 returns to step 420.

20 With reference still to Figure 4, in step 460, an eventOut is generated. In one embodiment, referring back to Figure 2, the corresponding time event of VideoTexture node 205 is returned via the eventOut named timeEvent and sent to script node 210 for further processing.

Referring again to Figure 4, in step 470, the appropriate changes in behavior for VRML objects in the scene as designated by the triggering time event are effectuated. In one embodiment, with reference back to Figure 2, script node 210 serves to launch one or more routes (e.g., routes 215) within the VRML scene to trigger the designated behavior changes of VRML objects. Upon the completion of step 470, process 400 returns to step 420. As thus described, event triggering in a VRML scene is implemented by using the VRML extensions (e.g., VideoTexture node 205 and VideoTexture extension 223) of the present invention.

#### Support for 3-D Content Authoring in accordance with the Present Invention

The present invention also provides a mechanism which facilitates the authoring of 3-D graphics content where events are triggered by a broadcast signal using a standard VRML authoring tool. More specifically, within the scope of the present invention, a content creator can develop and test 3-D content by utilizing a built-in event generation feature in VRML to simulate broadcast triggers. Once the development and testing has been completed, the content creator can then make simple modifications to the content based on the event flow such that live broadcast triggers can be processed when dynamic content is broadcast.

With reference next to Figure 5, a data flow diagram illustrating data flow for simulating event triggering in accordance with one embodiment of the present invention is shown. As illustrated in Figure 5, in a currently preferred embodiment, data is authored by a content creator and stored in a VRML scene

description file having a TimeCompare script node 530 of the present invention, a Trigger script node 540 as well as routes 550. Script nodes and routes are elements of VRML and are known to those of ordinary skill in the art. In one embodiment, a timelist of triggers is stored in the timelist field 530a of

5 TimeCompare script node 530.

Referring still to Figure 5, a Clock TimeSensor node 510 is used to generate simulation ticks on a periodic basis. In one embodiment, Clock TimeSensor node 510 includes a time eventOut for sending the corresponding  
10 time data 516 to a TimeConverter script node 520 upon each simulation tick. In one embodiment, TimeConverter script node 520 converts time data 516 that is the absolute time expressed in the VRML data type SFTTime to a traditional timecode quadruple 526, which is then sent to TimeCompare script node 530.

15 Within the scope of the present invention, TimeCompare script node 530 compares the content in timelist field 530a against timecode quadruple 526. When a match between a specified timestamp in the timelist and timecode quadruple 526 is detected, corresponding time event 538 is sent to Trigger script node 540 for further processing. In one embodiment, Trigger script node  
20 540 serves to launch one or more routes (e.g., routes 550) within the VRML scene when time event 538 is received from TimeCompare script node 530. Routes 550 in turn trigger behavior changes of VRML objects in the scene. It is appreciated that the content author is responsible for ensuring that there exists a one-to-one correspondence between the times listed in timelist field 530a of  
25 TimeCompare script node 530 and the time events listed in Trigger script node

540. Significantly, in this embodiment, the present invention allows a standard VRML authoring tool to be used for the authoring and editing of a VRML scene that can dynamically respond to broadcast triggers.

5       With reference next to Figure 6, a flow diagram illustrating steps for testing event triggering using simulated timecodes in accordance with one embodiment of the present invention is shown. In step 610, a timelist comprising video triggers is defined. In one embodiment, each of the video triggers represents a time at which an event is to occur within a VRML scene  
10       generated using the VRML platform of the present invention. In one embodiment, the timelist is stored in timelist field 530a of TimeCompare script node 530 of a VRML scene description file as depicted in Figure 5.

      With reference still to Figure 6, in step 615, a simulation tick is generated  
15       periodically as time elapses. In one embodiment, a Clock TimeSensor node generates the simulation tick. Moreover, in one embodiment, consecutive simulation ticks approximate real clock time.

      Referring still to Figure 6, in step 620, time data is sent from the Clock  
20       TimeSensor node to a TimeConverter script node whenever a simulation tick is generated. In one embodiment, the time data comprises the absolute time and is represented as VRML data type SFTIME.

      With reference still to Figure 6, in step 625, the time data received by the  
25       TimeConverter script node is converted to a traditional timecode quadruple. In

one embodiment, the timecode quadruple is represented as (h, m, s, f) wherein h stands for hour, m for minute, s for second, and f for frame.

Referring again to Figure 6, in step 630, the timecode quadruple is sent  
5 from the TimeConverter script node to a TimeCompare script node.

Referring still to Figure 6, in step 635, the timecode quadruple is compared against the timelist field of the TimeCompare script node. In one embodiment, the TimeCompare script node functions substantially the same as  
10 VideoTexture node 205 of Figure 2 as described above, except that the TimeCompare script node does not perform the task of enabling live video in a VRML scene. In this embodiment, a static image is used in place of a live broadcast video feed.

15 With reference again to Figure 6, in step 640, when a match is detected between one of the video triggers in the timelist and the timecode, a behavior change for an object in the 3-D graphics scene is effectuated according to the matched video trigger such that the behavior change is synchronized with the simulated video frame. Upon the completion of step 640, process 600 returns to  
20 step 615. In one embodiment, steps 615 through 640 can be repeated as many times as necessary until development of testing of the 3-D content (e.g., as specified in the VRML scene description file including the timelist of triggers) is completed. A method for testing event triggering in a VRML scene using simulated timecodes in accordance with an embodiment of the present  
25 invention is thus described.

Significantly, a content creator using the method for testing event triggering of the present invention as described above is able to create, test and edit 3-D graphics content that will dynamically respond to broadcast triggers simply by using a standard VRML authoring tool. Importantly, the content creator does not have to get directly involved with most of the underlying aspects of the present invention. More specifically, the content creator just needs to provide a list of timecodes indicating when event triggers should occur and associate each event trigger with certain defined action in the VRML scene. Given the timecodes and their associated actions as inputs, a standard VRML authoring tool can generate the necessary event structure, such as TimeSensor and script nodes (e.g., Clock TimeSensor node 510, TimeConverter script node 520, TimeCompare script node 530) and routes, for simulating broadcast trigger handling in accordance with the present invention. As such, the present invention provides a mechanism which facilitates the authoring of 3-D graphics content where events are triggered by a broadcast signal using a standard VRML authoring tool.

In one embodiment, the Clock TimeSensor node, the TimeConverter script node and the TimeCompare script node are collectively replaced by a VideoTexture node of the present invention, wherein the content of the timelist field of the VideoTexture script node is the same as the content of the TimeCompare script node as of the completion of content development. By so doing, live broadcast triggers can be used in place of the simulated triggers to achieve the same desired events in the VRML scene.



Dynamic Video Effects in accordance with the Present Invention

With reference back to Figure 2, within the scope of the present invention, rendering and compositing are performed in real-time in STBs 151, 152 and  
5 153 at the viewers' end, rather than during post-production prior to broadcast at the broadcaster's end. Significantly, by postponing the rendering and compositing phase until the program content reaches the viewer's end, broadcast programming can be highly personalized to cater to each individual viewer's desires. In particular, such late compositing enables the broadcaster  
10 to inject dynamic video effects into the integrated video and graphics scene.

In one embodiment of the present invention, an Ethernet is used as a delivery medium to inject real-time effects into the scene via special effects media streams. In this embodiment, a streamer head end capable of sending  
15 out a sequence of uncompressed RGBA images over the Ethernet is used as the video source. It is appreciated that the transmission data rate is dependent upon the frame size as well as the desired frame rate. In this regard, one embodiment of the present invention requires deterministic playback to guarantee a certain frame rate. It is further appreciated that the maximum IP  
20 packet size is a constraint to achieving the full frame rate of 30 Hz. As such, in one embodiment, a 32-frame sequence of 256x256 images is streamed over the Ethernet at a rate of 15 Hz. In this embodiment, each video frame is partitioned into multiple packets of smaller size for delivery. Upon receipt at the viewer's end (e.g., STBs 151, 152 and/or 153), the packets are reassembled  
25 into their respective frames.

In addition, within the scope of the present invention, compression technologies, such as those developed by the MPEG community, can be applied to embodiments of the present invention to improve the performance of streaming media delivery. Furthermore, other streaming technologies, such as the Synchronized Multimedia Integration Language (SMIL™) that has been recommended by the World Wide Web Consortium (W3C) for synchronizing multimedia streams, can also be used to implement dynamic video effects within the scope and spirit of the present invention.

#### Incorporating Multi-user Technology into the Present Invention

Within the scope of the present invention, blending shared 3-D virtual environments with TV broadcasting is enabled by multi-user technologies. In one embodiment, the Community Place architecture developed by Sony Corporation is the designated multi-user technology. In another embodiment, the emerging Core Living Worlds standard that is being developed for multi-user support in VRML can be used. It is appreciated that these and other multi-user technologies and/or protocols can be utilized to complete the loop of interaction between the viewer(s) and the broadcaster within the scope of the present invention wherein the control and authoring of content is shared. In one commercial version of a Community Place multi-user server (e.g., VES 180 of Figure 1B) based on a centralized client-server architecture, up to approximately 1000 simultaneous users can be supported. It is appreciated that a distributed Community Place multi-user server can be used within the scope of the present invention to provide scalability for supporting massive

multi-user applications. More specifically, in one embodiment, the distributed Community Place architecture comprises multiple servers (e.g., a plurality of VES 180 in the context of Figure 1B) working in conjunction with a consistency module that ensures global consistency within the virtual world supported by  
5 the various servers.

Moreover, within the present invention, the inherent high bandwidth of the DTV broadcast channel can be utilized as an additional path for sending information from the servers to the clients to further enhance scalability. In one  
10 embodiment, the virtual world supported by the servers has two levels of information updates. The first level of information updates involves "piggy-backing" of updates over a high data rate DTV broadcast channel and is limited to sending global updates of the virtual world. The second level of information updates uses a different communication link, such as the Internet, to transmit  
15 local updates to a subset of viewers. In addition, rendering tasks within the virtual world can be partitioned into upstream and downstream components, which, in one embodiment, are performed at the head end and the viewer's end, respectively. For example, in a multi-player game or game show, elements of the background are rendered at the head end (e.g., by the server; by the  
20 broadcaster prior to broadcasting) and then broadcast to clients as an environment map. Client-specific elements are rendered at the viewers' end. As such, use of the available bandwidth can be optimized. In another embodiment, the DTV broadcast channel is used for both the "piggy-backing" of updates and shared rendering described above. In yet another embodiment,

load balancing and distribution among multiple servers is used to further optimize performance.

### INTERACTIVE MUSIC VIDEO

#### 5           IN ACCORDANCE WITH THE PRESENT INVENTION

One currently preferred embodiment of the present invention combines the broadcast stream by which traditional music video content is delivered with a 3-D graphics environment that gives the viewer an additional dimension of control and interaction with the video content. In this embodiment, the setting of  
10 a "virtual concert hall" is used. Importantly, in accordance with the present invention, the viewer is able to freely navigate in the 3-D environment and interact with objects in the scene representing the virtual concert hall. In one embodiment, the stage of the virtual concert hall features an animated 3-D model of a band, behind which is a screen showing a video feed of the band's  
15 recorded performance. In one embodiment, the motion of the 3-D band model is based on the live performance.

Moreover, with reference back to Figure 2, in one embodiment, using timecode 226 from video source 240, the music video of the instant embodiment  
20 can trigger events in the 3-D graphics scene. In an exemplary embodiment, triggers can be set up such that during the course of the music video, whenever the band sings a particular phrase of a song (e.g., reprise or chorus) the graphical spotlights in the 3-D scene will strobe. Triggers can also be set to effectuate changes in camera viewpoint as well as opening and/or closing of  
25 the stage curtain. In one embodiment, triggers are set to cause the curtain to

open and the band to begin playing at a designated animation start time, and to cause the curtain to close and the band to end its performance at a designated animation stop time. In addition, triggers can be set to activate and deactivate the strobing of the spotlights during the performance.

5

In a currently preferred embodiment, event triggering is also used to provide synchronization between multiple media streams, such as a video stream and an animation stream. In particular, if the viewer pauses the video, the animation is automatically paused as well. When the video resumes  
10 playing, the animation promptly continues from the point where it left off. Moreover, in this embodiment, such event triggering is implemented according to the process described above with reference to Figures 2 and 3.

Furthermore, in one embodiment of the present invention, when the  
15 viewer selects a passive viewing mode, the camera triggering events are processed as they are received from the video source and the camera automatically moves to the broadcaster's recommended view according to the triggers as the music video plays. Additionally, in one embodiment, special effect signals (e.g., special effect media streams) are transmitted to provide  
20 shaped video footage (e.g., falling leaves, snow) as described above. More particularly, in one embodiment, an effect layer is superimposed across the stage in a zigzag fashion to provide a sense of depth. Since these effects are partially transparent as described previously with respect to shaped video, such an effect layer seamlessly integrates with the rest of the scene. Moreover, the  
25 broadcaster has the freedom and flexibility to change the effects at any time.

In another embodiment, shaped video is used to introduce characters as video in a "virtual set" (e.g., video footage of narrators being inserted into a 3-D scene) that is controlled by the viewer. It is appreciated that while virtual set  
5 technology has been used as a production technique, it has not been exploited downstream in the device where viewers view the contents (e.g., STBs 151, 152 and 153 of Figure 1B). As consumers become increasingly accustomed to navigating and manipulating 3-D user interfaces, such as those presented by existing game consoles like the Sony PlayStation™, the interactive control of  
10 virtual sets of the present invention as described above will become an intuitive mechanism for viewers to interact with broadcast content, thereby providing a variety of media experiences that are not available in tradition TV viewing.

Thus, shared control of the overall media experience between the  
15 broadcaster and the viewer is made possible by the present invention. For instance, the broadcaster can present imagery to viewers over the broadcast channel and each viewer can selectively view, control or manage the imagery as if it was local content in a 3-D graphics environment. In particular, such possibilities add a profound new dimension to music videos since viewers can  
20 manipulate 3-D content to create their own custom experiences. Moreover, by exploiting the interface between the DTV data channel and the 3-D graphics scene, the broadcaster can simplify the process by which viewers can explore and author a rich set of media events with guaranteed synchronization to the broadcast video content.

Although certain embodiments of the present invention as described herein pertains to interactive music video, it is appreciated that many other advantageous applications are possible within the scope of the present invention. For example, the present invention can be advantageously applied to augment networked multi-player games, such as role-playing and/or strategy games, so that broadcast video can be used to add a live element and enhance fidelity of the games. In another embodiment, the present invention can be advantageously used in inhabited motion pictures, wherein viewers can explore virtual versions of movies with broadcast appearances by celebrities. Furthermore, in yet another embodiment, the present invention can be advantageously used to set up virtual museum exhibitions, wherein video and interactive graphics can be combined to make the television a surrogate museum. Therefore, the present invention enables numerous possibilities in a new broadcast paradigm wherein the broadcaster and the viewer can share control of the media content. It should be clear to a person of ordinary skill in the art, having read the description of embodiments of the present invention herein, that other applications and embodiments not expressly described herein are also possible without departing from the scope of the present invention.

20

The preferred embodiment of the present invention, a system and method for interfacing 3-D graphics content with broadcast video to generate interactive media content wherein the broadcaster and the viewer can share control of the media content, is thus described. While the present invention has been described in particular embodiments, it should be appreciated that the

25

broadcast video referred to herein is merely an example of an independent source of video information, and that any such independent source of video information, such as from video tapes, DVDs, or cable, etc., would function identically.

Moreover, one skilled in the art will realize that the techniques taught herein for

5 interaction between graphics and video are not restricted to 3-D graphics, but would work with two-dimensional graphics, and indeed, since time information is included, the invention could be practiced in a four-dimensional graphics development.

10 In general, the present invention should not be construed as limited to the particular embodiments described herein, but rather construed according to the following claims.



CLAIMS

What is claimed is:

- 5           1.     A computer implemented method for interfacing a three-dimensional (3-D) graphics platform with broadcast video, said method comprising the steps of:
- a)     defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a 3-D graphics scene generated using said 3-D graphics platform;
- 10           b)     accessing said timelist from a data file, said data file being an input format supported by said 3-D graphics platform;
- c)     receiving a timecode and a video frame from said broadcast video, said timecode being associated with said video frame;
- d)     comparing said video triggers and said timecode; and
- 15           e)     responsive to a match between one of said video triggers and said timecode, effectuating a behavior change for an object in said 3-D graphics scene such that said behavior change is synchronized with said video frame in real-time.
2.     The method as recited in Claim 1 wherein said 3-D graphics platform
- 20           comprises a Virtual Reality Modeling Language (VRML) platform and wherein said data file comprises a VRML scene description file.
3.     The method as recited in Claim 2 wherein said timelist is stored in a VRML node of said VRML scene description file.
- 25           4.     The method as recited in Claim 1 further comprising the step of embedding shaped video in said broadcast video, said shaped video being partially transparent to provide special effects generated within said 3-D graphics scene.

5. The method as recited in Claim 1 further comprising the step of accepting input from a viewer such that contents of said broadcast video are capable of changing in response to said input.

5 6. The method as recited in Claim 1 further comprising the step of accepting input from a viewer such that said input is capable of effectuating said behavior change for said object of said 3-D graphics scene.

7. The method as recited in Claim 1 further comprising the steps of  
10 transmitting feedback information to a source of said broadcast video and said source modifying contents of said broadcast video in response to said feedback information.

8. The method as recited in Claim 7 wherein said feedback information  
15 is transmitted over the Internet.

9. The method as recited in Claim 1 wherein said broadcast video is transmitted over a digital television (DTV) data channel.

20 10. The method as recited in Claim 1 wherein said broadcast video comprises music video.

11. A computer system comprising a processor coupled to a bus and a memory unit coupled to said bus, said memory unit having stored therein  
25 instructions that when executed implement a method of interfacing a three-dimensional (3-D) graphics platform with broadcast video, said method comprising the steps of:

a) defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a 3-D graphics  
30 scene generated using said 3-D graphics platform;

b) accessing said timelist from a data file, said data file being an input format supported by said 3-D graphics platform;

c) receiving a timecode and a video frame from said broadcast video, said timecode being associated with said video frame;

5 d) comparing said video triggers and said timecode; and

e) responsive to a match between one of said video triggers and said timecode, effectuating a behavior change for an object in said 3-D graphics scene such that said behavior change is synchronized with said video frame in real-time.

10 12. The computer system as recited in Claim 11 wherein said 3-D graphics platform comprises a Virtual Reality Modeling Language (VRML) platform and wherein said data file comprises a VRML scene description file.

13. The computer system as recited in Claim 12 wherein said timelist is  
15 stored in a VRML node of said VRML scene description file.

14. The computer system as recited in Claim 11 wherein said method further comprises the step of embedding shaped video in said broadcast video, said shaped video being partially transparent to provide special effects generated  
20 within said 3-D graphics scene.

15. The computer system as recited in Claim 11 wherein said method further comprises the step of accepting input from a viewer such that contents of said broadcast video are capable of changing in response to said input.  
25

16. The computer system as recited in Claim 11 wherein said method further comprises the step of accepting input from a viewer such that said input is capable of effectuating said behavior change for said object of said 3-D graphics scene.  
30

17. The computer system as recited in Claim 11 wherein said method further comprises the steps of transmitting feedback information to a source of said broadcast video and said source modifying contents of said broadcast video in response to said feedback information.

5

18. The computer system as recited in Claim 17 wherein said feedback information is transmitted over the Internet.

19. The computer system as recited in Claim 11 wherein said broadcast  
10 video is transmitted over a digital television (DTV) data channel.

20. The computer system as recited in Claim 11 wherein said broadcast video comprises music video.

15 21. A computer system for interfacing a three-dimensional (3-D) graphics platform with broadcast video, said computer system comprising:

authoring means for defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a 3-D graphics scene generated using said 3-D graphics platform;

20 inputting means for accessing said timelist from a data file, said data file being an input format supported by said 3-D graphics platform;

said inputting means also for receiving a timecode and a video frame from said broadcast video, said timecode being associated with said video frame;

comparing means for comparing said video triggers and said timecode; and

25 triggering means for effectuating a behavior change for an object in said 3-D graphics scene in response to a match between one of said video triggers and said timecode such that said behavior change is synchronized with said video frame in real-time.

30 22. The computer system as recited in Claim 21 wherein said 3-D

graphics platform comprises a Virtual Reality Modeling Language (VRML) platform and wherein said data file comprises a VRML scene description file.

23. The computer system as recited in Claim 22 wherein said timelist is  
5 stored in a VRML node of said VRML scene description file.

24. The computer system as recited in Claim 21 wherein said broadcast video comprises shaped video, said shaped video being partially transparent to provide special effects generated within said 3-D graphics scene.

10

25. The computer system as recited in Claim 21 further comprising transmitting means for transmitting feedback information to a source of said broadcast video wherein said source modifies contents of said broadcast video in response to said feedback information.

15

26. The computer system as recited in Claim 25 wherein said feedback information is transmitted over the Internet.

27. The computer system as recited in Claim 21 wherein said broadcast  
20 video is transmitted over a digital television (DTV) data channel.

28. The computer system as recited in Claim 21 wherein said broadcast video comprises music video.

25 29. A computer implemented method for interfacing a three-dimensional (3-D) graphics platform with independent video information, said method comprising the steps of:

a) defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a 3-D graphics  
30 scene generated using said 3-D graphics platform;

b) receiving a timecode and a video frame from said independent video information, said timecode being associated with said video frame;

c) comparing said video triggers and said timecode; and

d) effectuating a behavior change for an object in said 3-D graphics scene based upon a relationship between one of said video triggers and said timecode such that said behavior change is synchronized with said video frame.

30. A computer system comprising a processor coupled to a bus and a memory unit coupled to said bus, said memory unit having stored therein instructions that when executed implement a method of interfacing a graphics platform with independent video, said method comprising the steps of:

a) defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a scene generated using said graphics platform;

b) receiving a timecode and a video frame from said independent video, said timecode being associated with said video frame;

c) comparing said video triggers and said timecode; and

d) responsive to a match between one of said video triggers and said timecode, effectuating a change in said graphics scene such that said change in scene is synchronized with said video frame.

31. A computer system for interfacing a graphics platform with independent video, said computer system comprising:

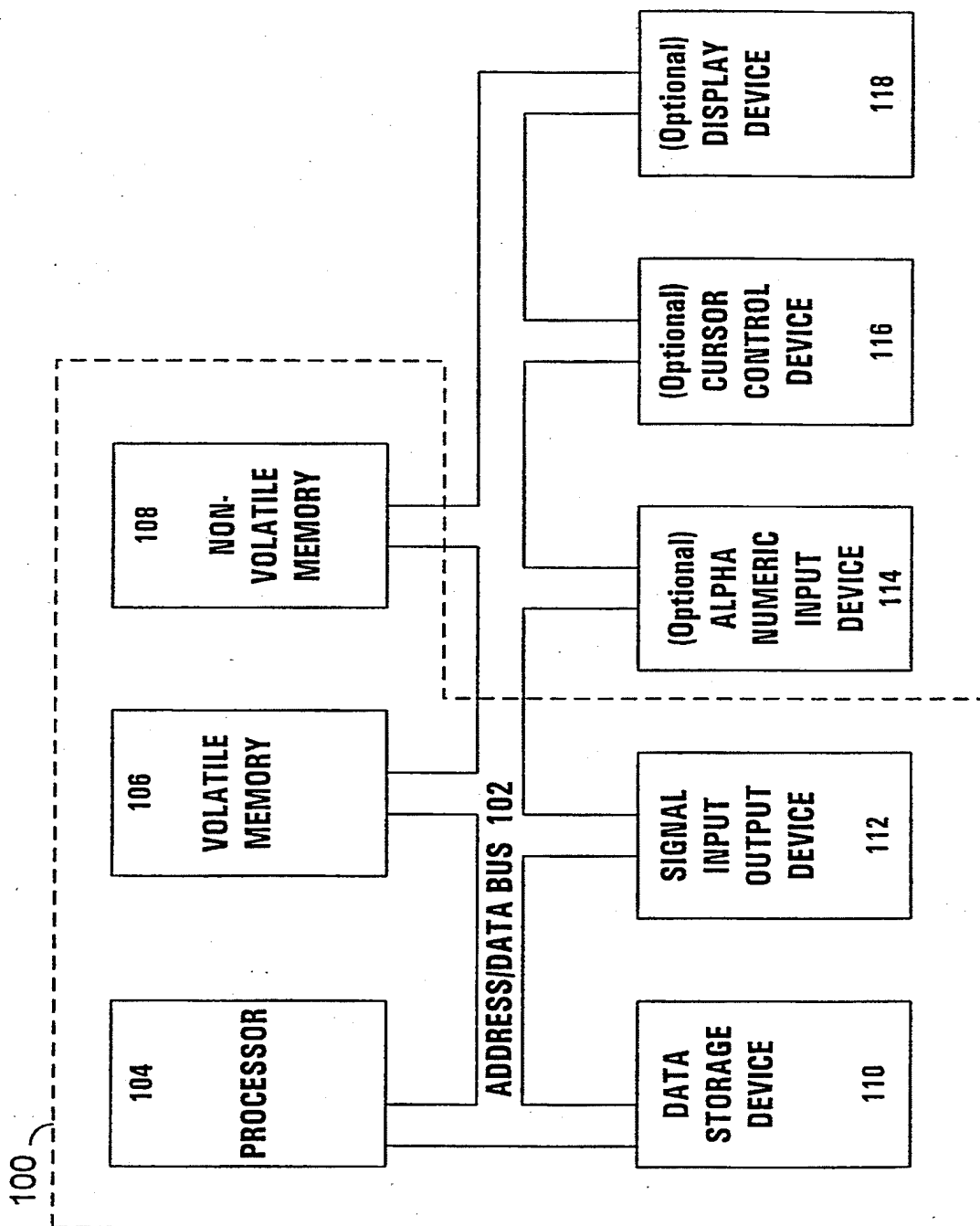
authoring means for defining a timelist comprising video triggers, each of said video triggers representing a time at which an event is to occur within a graphics scene generated using said graphics platform;

inputting means for receiving a timecode and a video frame from said independent video, said timecode being associated with said video frame;

comparing means for comparing said video triggers and said timecode; and

triggering means for effectuating a behavior change for an object in said

graphics scene in response to a relationship between one of said video triggers and said timecode such that said behavior change is synchronized with said video frame.



**FIGURE 1A**



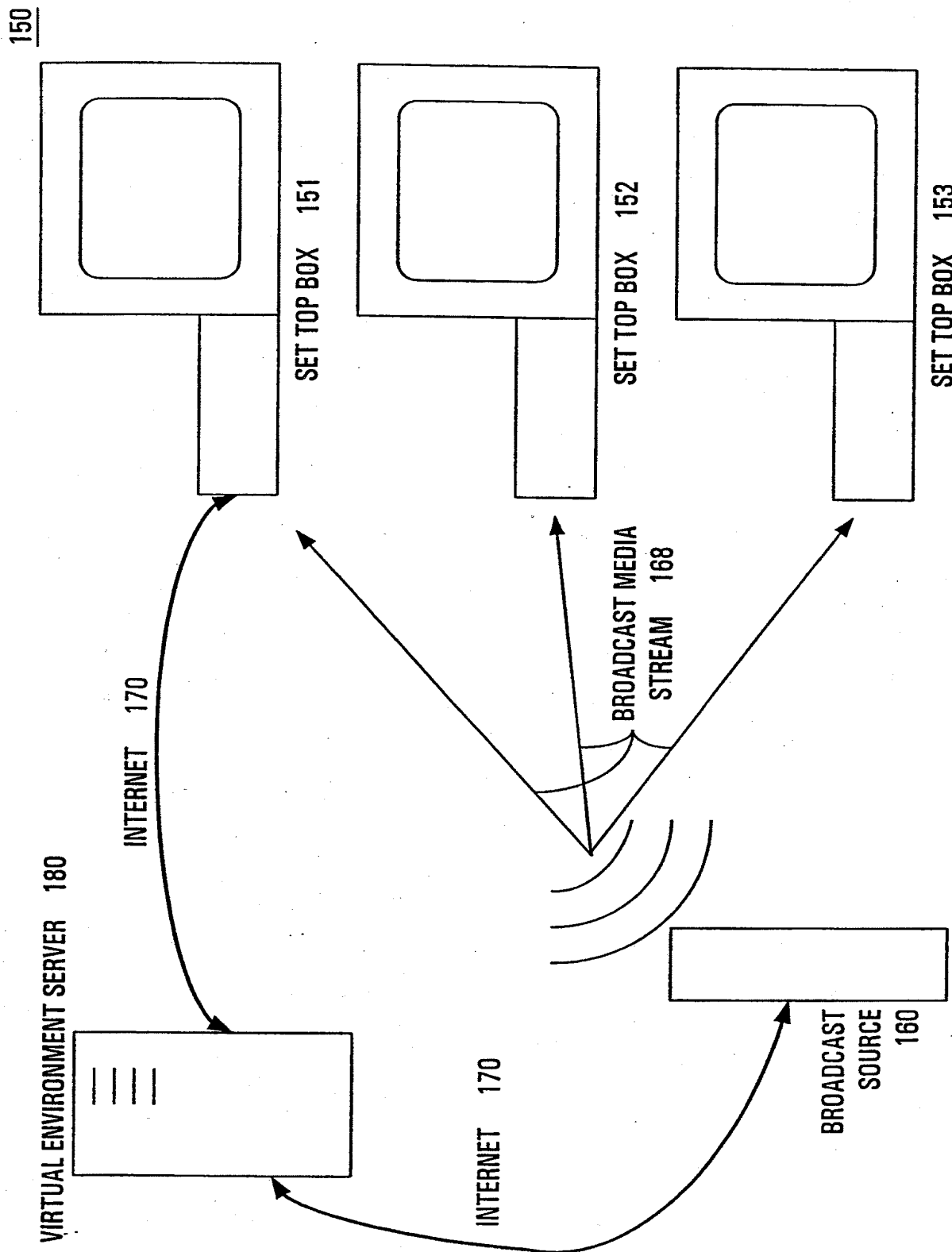


FIGURE 1B

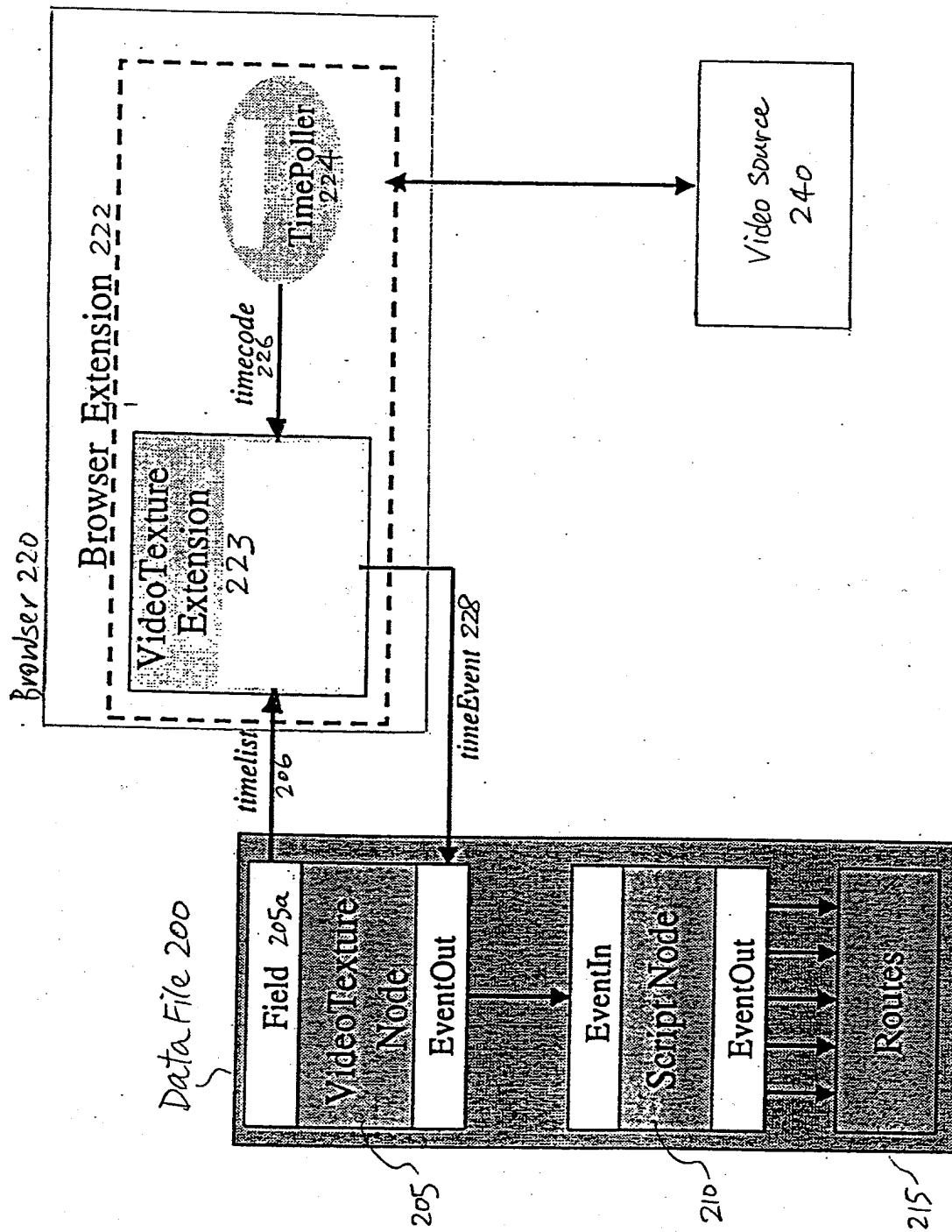
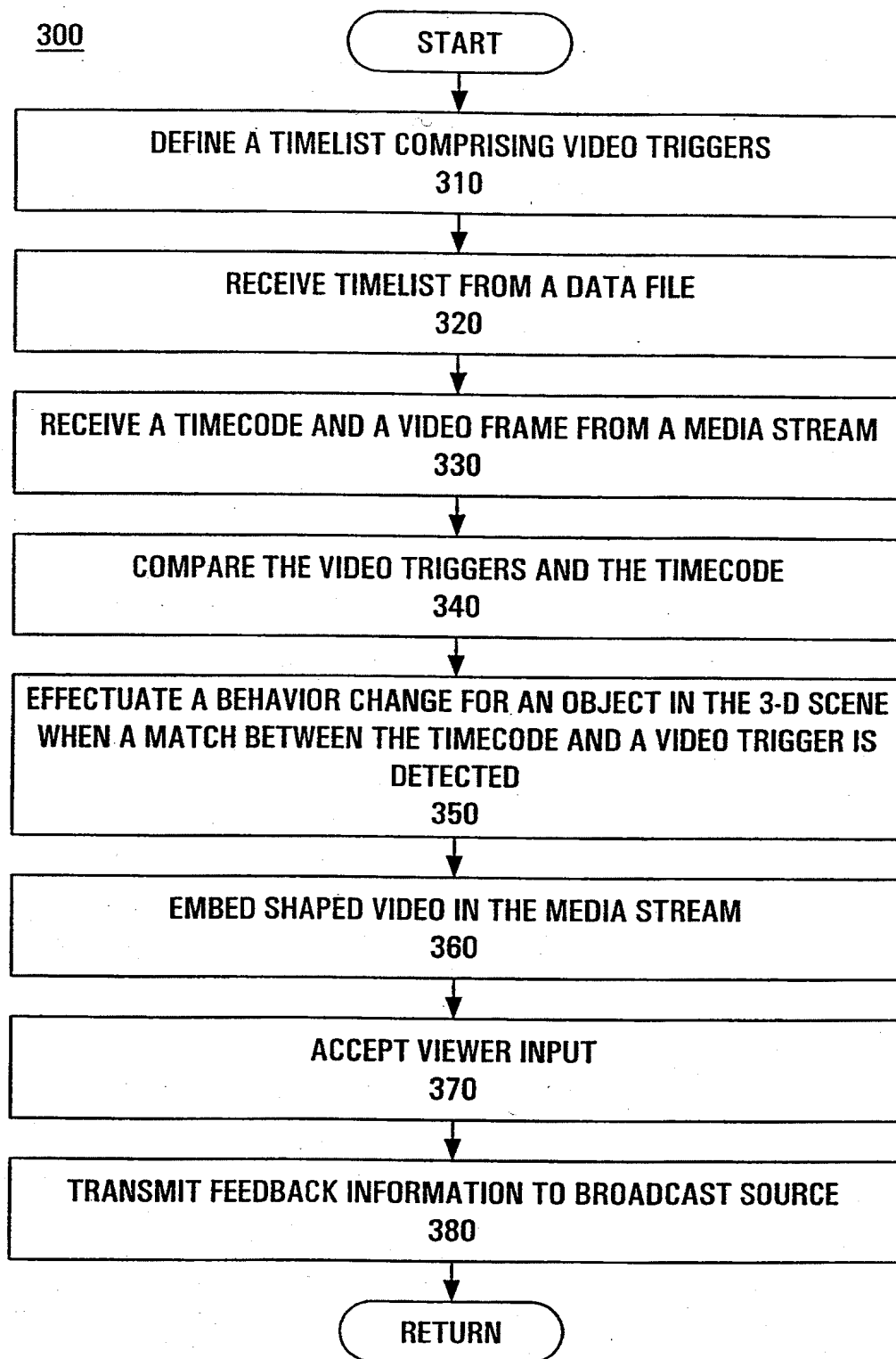
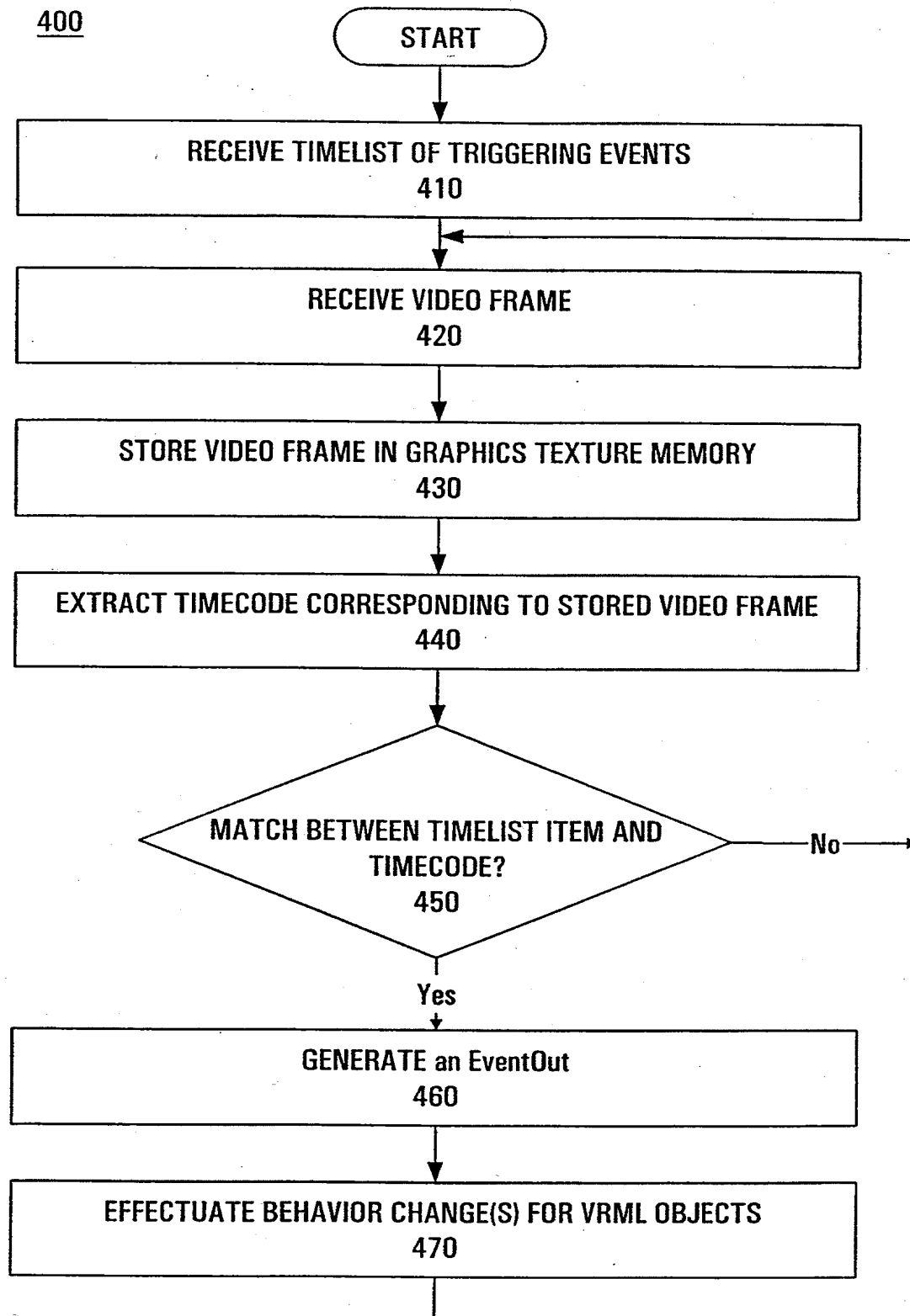


Figure 2

**FIGURE 3**

**FIGURE 4**

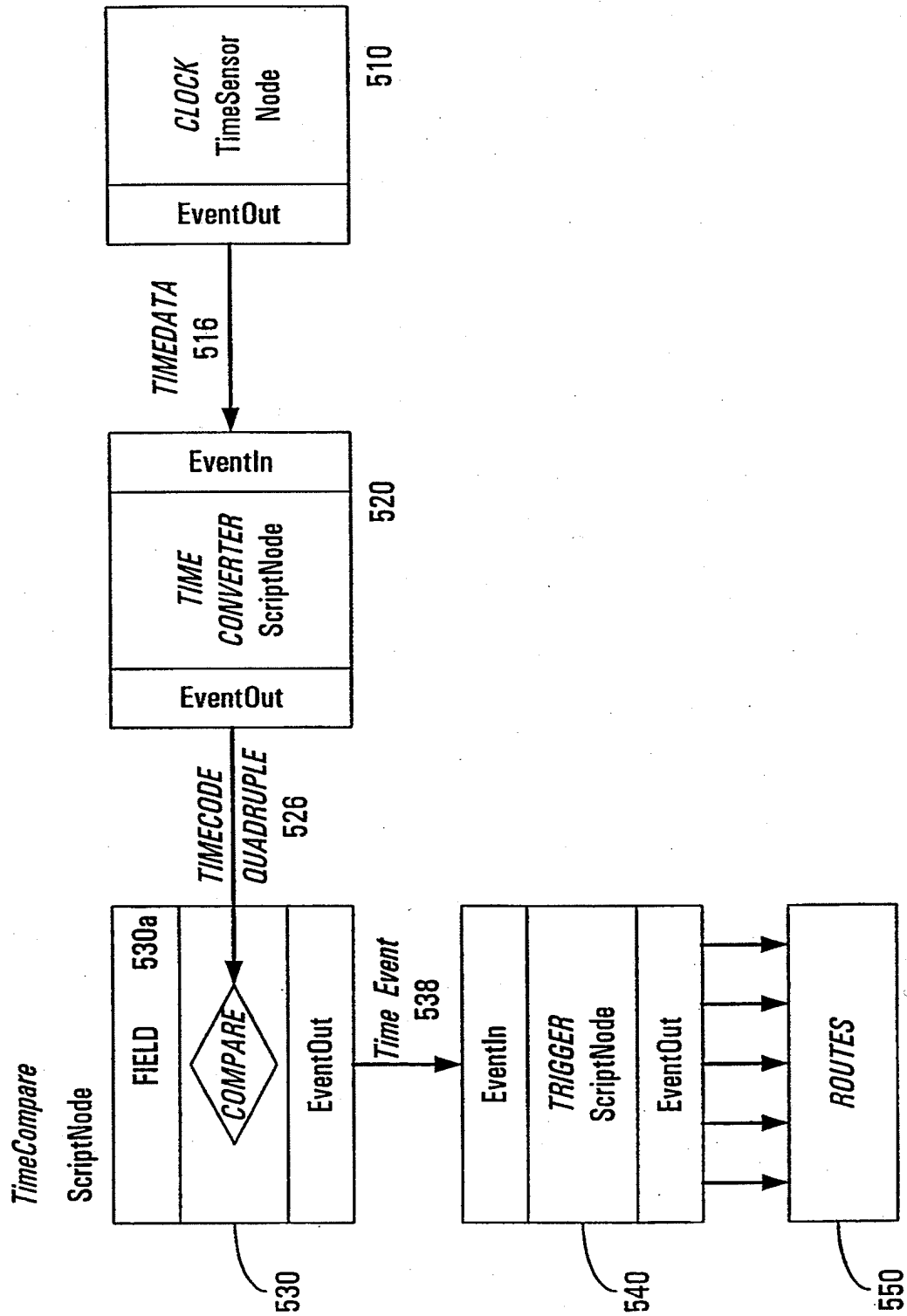
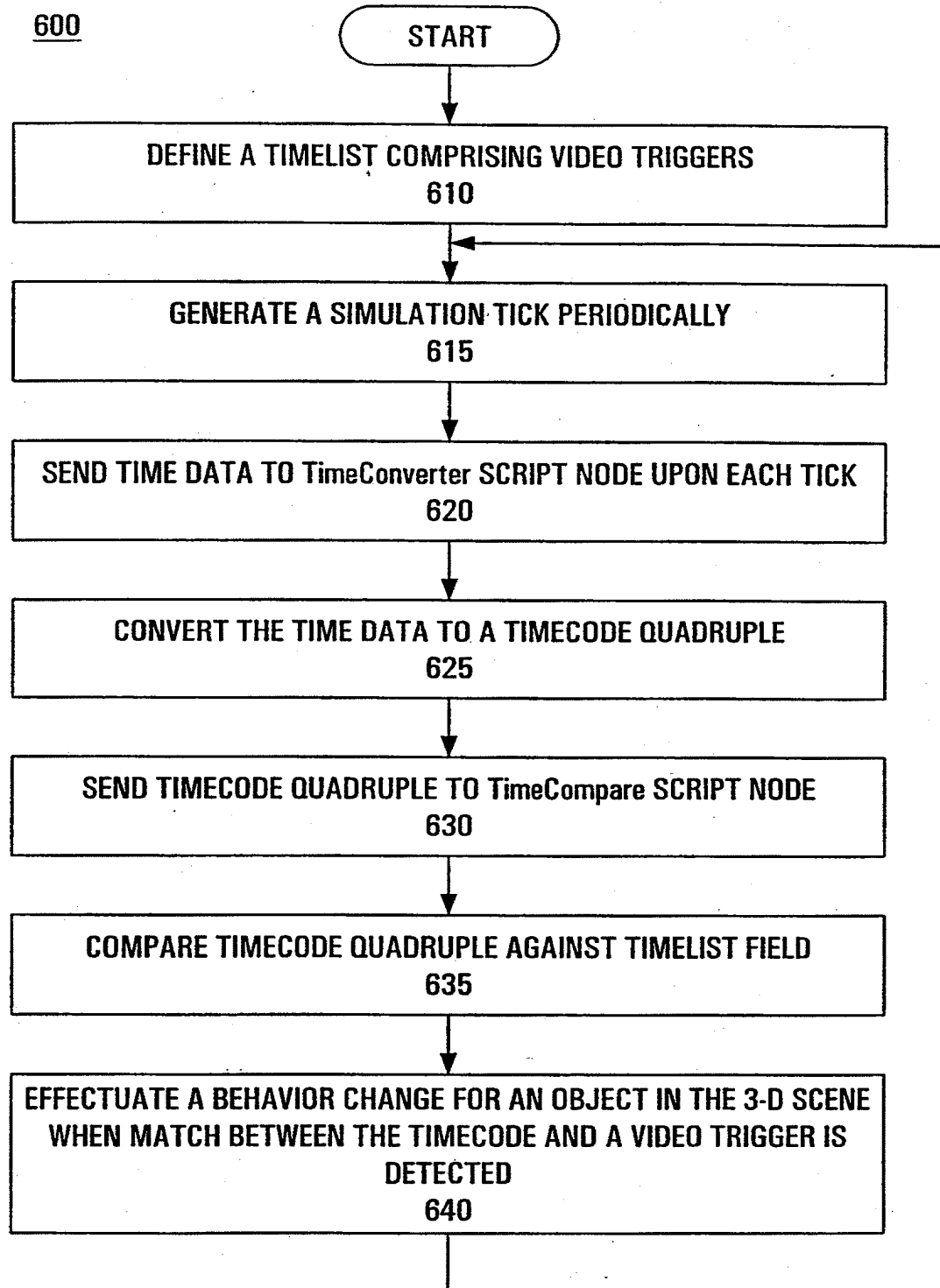


FIGURE 5

**FIGURE 6**

# INTERNATIONAL SEARCH REPORT

National Application No  
PCT/US 00/01265

**A. CLASSIFICATION OF SUBJECT MATTER**  
IPC 7 H04N7/26 H04N7/173

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)  
IPC 7 H04N G06T

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	<p>AVARO O ET AL: "The MPEG-4 systems and description languages: A way ahead in audio visual information representation" SIGNAL PROCESSING. IMAGE COMMUNICATION, NL, ELSEVIER SCIENCE PUBLISHERS, AMSTERDAM, vol. 9, no. 4, 1 May 1997 (1997-05-01), pages 385-431, XP004075337 ISSN: 0923-5965 page 397, line 11 - line 17</p> <p style="text-align: center;">--- -/-</p>	<p>1,2,4,9, 11,12, 14,19, 21,22, 24,27, 29-31</p>

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

\* Special categories of cited documents :

- \*A\* document defining the general state of the art which is not considered to be of particular relevance
- \*E\* earlier document but published on or after the international filing date
- \*L\* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- \*O\* document referring to an oral disclosure, use, exhibition or other means
- \*P\* document published prior to the international filing date but later than the priority date claimed

- \*T\* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- \*X\* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- \*Y\* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- \*G\* document member of the same patent family

Date of the actual completion of the international search

26 May 2000

Date of mailing of the international search report

05/06/2000

Name and mailing address of the ISA  
European Patent Office, P.B. 5818 Patentlaan 2  
NL - 2280 HV Rijswijk  
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,  
Fax: (+31-70) 340-3016

Authorized officer

Sindic, G

## INTERNATIONAL SEARCH REPORT

national Application No  
PCT/US 00/01265

## C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	"International Standard ISO/IEC 14772-1: 1997 Virtual Reality Modeling Language (VRML97)" 'ONLINE!', 1997, pages 1-236, XP002133320 Retrieved from the Internet: <URL:www.vrml.org/Specifications/VRML97> 'retrieved on 2000-03-02! page 52 -page 55 page 102 -page 103 -----	1-3,6, 11-13, 16, 21-23, 29-31
P,X	EP 0 924 934 A (NIPPON ELECTRIC CO) 23 June 1999 (1999-06-23)  column 14, line 13 - line 32 column 18, line 58 -column 19, line 9 -----	1-3,9, 11-13, 19, 21-23, 27,29-31



# INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 00/01265

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
---	---------------------	----------------------------	---------------------

EP 0924934 A	23-06-1999	JP 11187398 A	09-07-1999
--------------	------------	---------------	------------